

What is claimed is:

1. In a computer system having a scalar processing unit and a vector processing unit, wherein the vector processing unit includes a vector dispatch unit, a method of executing a vector memory instruction having a scalar operand, the method comprising:
 - reading the scalar operand, wherein reading includes transferring the scalar operand from the scalar processing unit to the vector dispatch unit;
 - determining if the vector memory instruction is scalar committed; and
 - if the vector memory instruction is scalar committed, executing the vector memory operation as a function of the scalar operand.
2. The method according to claim 1, wherein executing the vector memory operation includes translating an address associated with the vector memory operation and trapping on a translation fault.
3. In a computer system having a scalar processing unit and a vector processing unit, a method of decoupling vector data loads from vector instruction execution, comprising:
 - generating an address for a vector load;
 - issuing a vector load request to memory;
 - receiving vector data from memory;
 - storing the vector data in a load buffer;
 - transferring the vector data from the load buffer to a vector register; and
 - executing a vector instruction on the vector data stored in the vector register.
4. The method according to claim 3, wherein the vector processing unit includes a vector execute unit and a vector load/store unit, wherein issuing a vector load request to memory includes issuing and executing vector memory references in the vector load/store unit when the vector load store unit has received the instruction and memory operands from the scalar processing unit.

5. In a computer system having a scalar processing unit and a vector processing unit, a method of decoupling vector data loads from vector instruction execution, comprising:

- generating a first and a second address for a vector load;
- issuing first and second vector load requests to memory;
- receiving vector data associated with the first and second addresses from memory;
- storing vector data associated with the first address in a first vector register;
- storing vector data associated with the second address in a second vector register;
- executing a vector instruction on the vector data stored in the first vector register;
- renaming the second vector register; and
- executing a vector instruction on the vector data stored in the second vector register.

6. The method according to claim 3, wherein the vector processing unit includes a vector execute unit and a vector load/store unit, wherein issuing a vector load request to memory includes issuing and executing vector memory references in the vector load/store unit when the vector load store unit has received the instruction and memory operands from the scalar processing unit.

7. A computer system, comprising:

- a scalar processing unit; and
- a vector processing unit, wherein the vector processing unit includes a vector execute unit and a vector load/store unit;
 - wherein the vector load/store unit receives an instruction and memory operands from the scalar processing unit, issues and executes a vector memory load reference as a function of the instruction and the memory operands received from the scalar processing

unit, and stores data received as a result of the vector memory reference in a load buffer;
and

wherein the vector execute unit issues the vector memory load instruction and transfers the data received as a result of the vector memory reference from the load buffer to a vector register.

8. In a computer system having a scalar processing unit and a vector processing unit, a method of decoupling scalar and vector execution, comprising:

dispatching scalar instructions to a scalar instruction queue;

dispatching a vector instruction that requires scalar operands to the scalar instruction queue and to a vector instruction queue;

executing the vector instruction in the scalar processing unit, wherein executing the vector instruction in the scalar processing unit includes writing a scalar operand to a scalar operand queue;

notifying the vector processing unit that the scalar operand is available in the scalar operand queue; and

executing the vector instruction in the vector processing unit, wherein executing the vector instruction in the vector processing unit includes reading the scalar operand from the scalar operand queue.

9. In a computer system having a scalar processing unit and a vector processing unit, a method of generating an address for a vector memory reference, comprising:

dispatching scalar instructions to a scalar instruction queue;

dispatching a vector instruction to the scalar instruction queue and to a vector instruction queue;

executing the vector instruction in the scalar processing unit, wherein executing the vector instruction in the scalar processing unit includes generating an address and writing the address to a scalar operand queue;

notifying the vector processing unit that the address is available in the scalar operand queue; and

executing the vector instruction in the vector processing unit, wherein executing the vector instruction in the vector processing unit includes reading the address from the scalar operand queue and generating a memory request as a function of the address read from the scalar operand queue.

10. In a computer system having a scalar processing unit and a vector processing unit, a method of executing a vector instruction, comprising:

dispatching scalar instructions to a scalar instruction queue;

dispatching a vector instruction to the scalar instruction queue and to a vector instruction queue;

executing the vector instruction in the scalar processing unit, wherein executing the vector instruction in the scalar processing unit includes generating an address and writing the address to a scalar operand queue;

notifying the vector processing unit that the address is available in the scalar operand queue; and

executing the vector instruction in the vector processing unit, wherein executing the vector instruction in the vector processing unit includes:

reading the address from the scalar operand queue;

generating a memory request as a function of the address read from the scalar operand queue;

receiving vector data from memory;

storing the vector data in a load buffer;

transferring the vector data from the load buffer to a vector register; and

executing a vector instruction on the vector data stored in the vector register.

11. In a computer system having a scalar processing unit and a vector processing unit, a method of unrolling a loop, comprising:

preparing a first and a second vector instruction, wherein each vector instruction execute an iteration through the loop and wherein each vector instruction requires calculation of a scalar loop value;

dispatching the first and second vector instructions to the scalar instruction queue and to a vector instruction queue;

executing each vector instruction in the scalar processing unit, wherein executing each vector instruction in the scalar processing unit includes writing a scalar operand representing the scalar loop value calculated for each vector instruction to a scalar operand queue;

notifying the vector processing unit that the scalar operand is available in the scalar operand queue;

executing the first and second vector instructions in the vector processing unit, wherein executing the vector instruction in the vector processing unit includes reading the scalar operands associated with each instruction from the scalar operand queue.